Laborator si seminar
Programare in Java si software matematic

Adrian Ichimescu

Adrianichimescu@gmail.com

Recap: Classes and Objects

- **Object** – Objects have states (variables) and behaviors (methods)
- **Class** – A class can be defined as a template that describes the behavior/state that the object of its type support.
  - **Local variables** – Variables defined inside methods, constructors or blocks are called local variables.
  - **Instance variables** – Instance variables are variables within a class but outside any method.
  - **Class variables -** Class variables are variables declared within a class, outside any method, with the static keyword.

Interface

A class describes the attributes and behaviors of an object. An interface contains behaviors that a class implements.

Unless the class that implements the interface is abstract, all the methods of the interface need to be defined in the class.

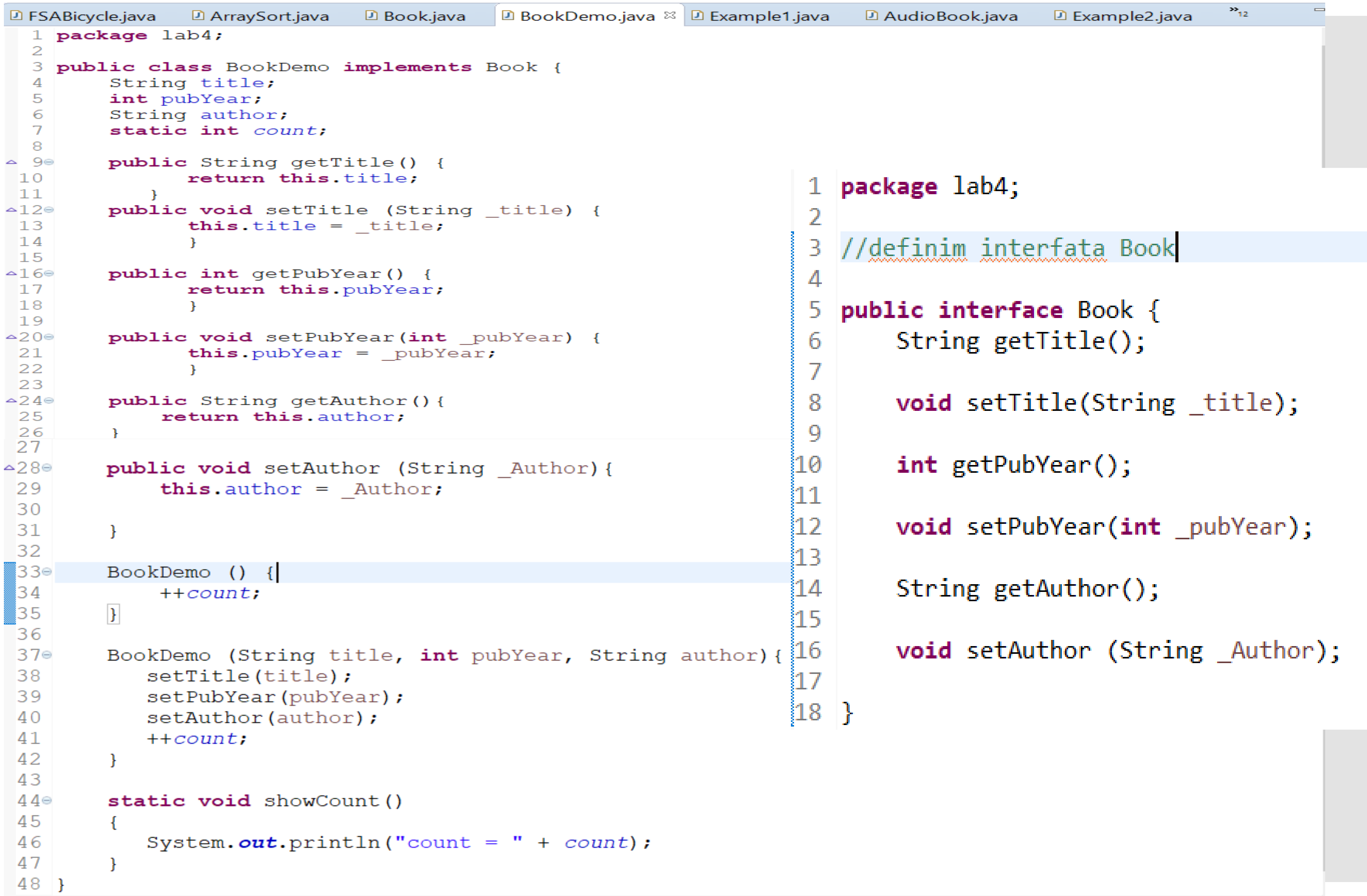
An interface is similar to a class in the following ways

- An interface can contain any number of methods.
- An interface is written in a file with a **.java** extension, with the name of the interface matching the name of the file.
- The byte code of an interface appears in a **.class** file.
- Interfaces appear in packages, and their corresponding bytecode file must be in a directory structure that matches the package name.

An interface is different from a class in several ways, including –

- You cannot instantiate an interface.
- An interface does not contain any constructors.
- All of the methods in an interface are abstract.
- An interface cannot contain instance fields. The only fields that can appear in an interface must be declared both static and final.
- An interface is not extended by a class; it is implemented by a class.
- A class can implement multiple interfaces.

Interface



```java
package lab4;

public class BookDemo implements Book {
	String title;
	int pubYear;
	String author;
	static int count;

	public String getTitle() {
			return this.title;
		}
	public void setTitle (String _title) {
			this.title = _title;
			}

	public int getPubYear() {
			return this.pubYear;
			}

	public void setPubYear(int _pubYear) {
			this.pubYear = _pubYear;
			}

	public String getAuthor(){
		return this.author;
	}

	public void setAuthor (String _Author){
		this.author = _Author;

	}

	BookDemo () {
		++count;
	}

	BookDemo (String title, int pubYear, String author){
		setTitle(title);
		setPubYear(pubYear);
		setAuthor(author);
		++count;
	}

	static void showCount()
	{
		System.out.println("count = " + count);
	}
}
```

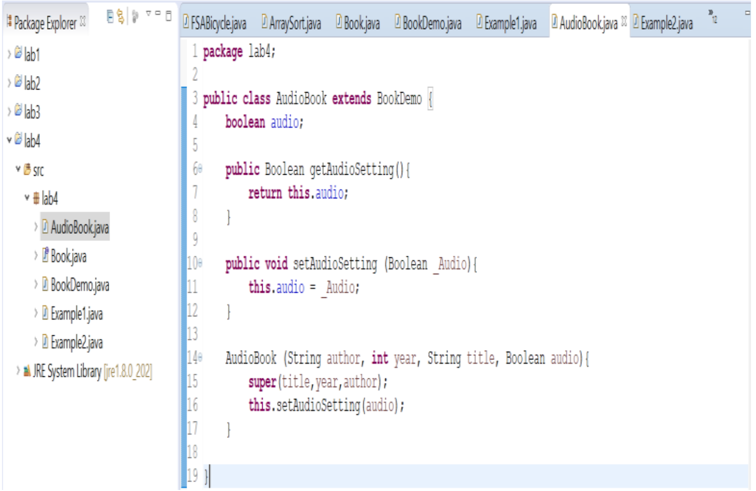
```java
package lab4;

//definim interfata Book

public interface Book {
	String getTitle();

	void setTitle(String _title);

	int getPubYear();

	void setPubYear(int _pubYear);

	String getAuthor();

	void setAuthor (String _Author);

}
```

Class Inheritance

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.

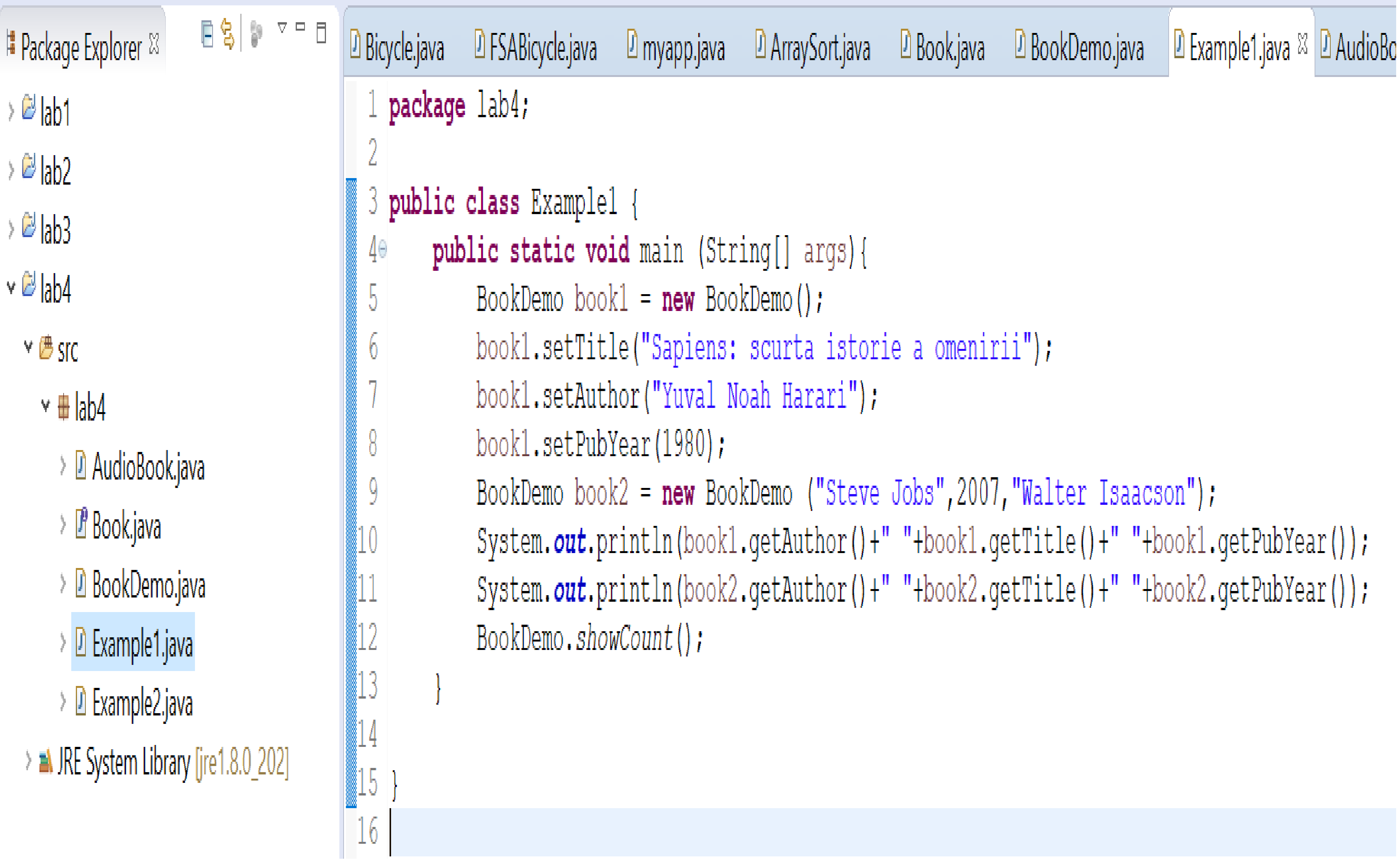
The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).

extends Keyword: **extends** is the keyword used to inherit the properties of a class. Following is the syntax of extends keyword.



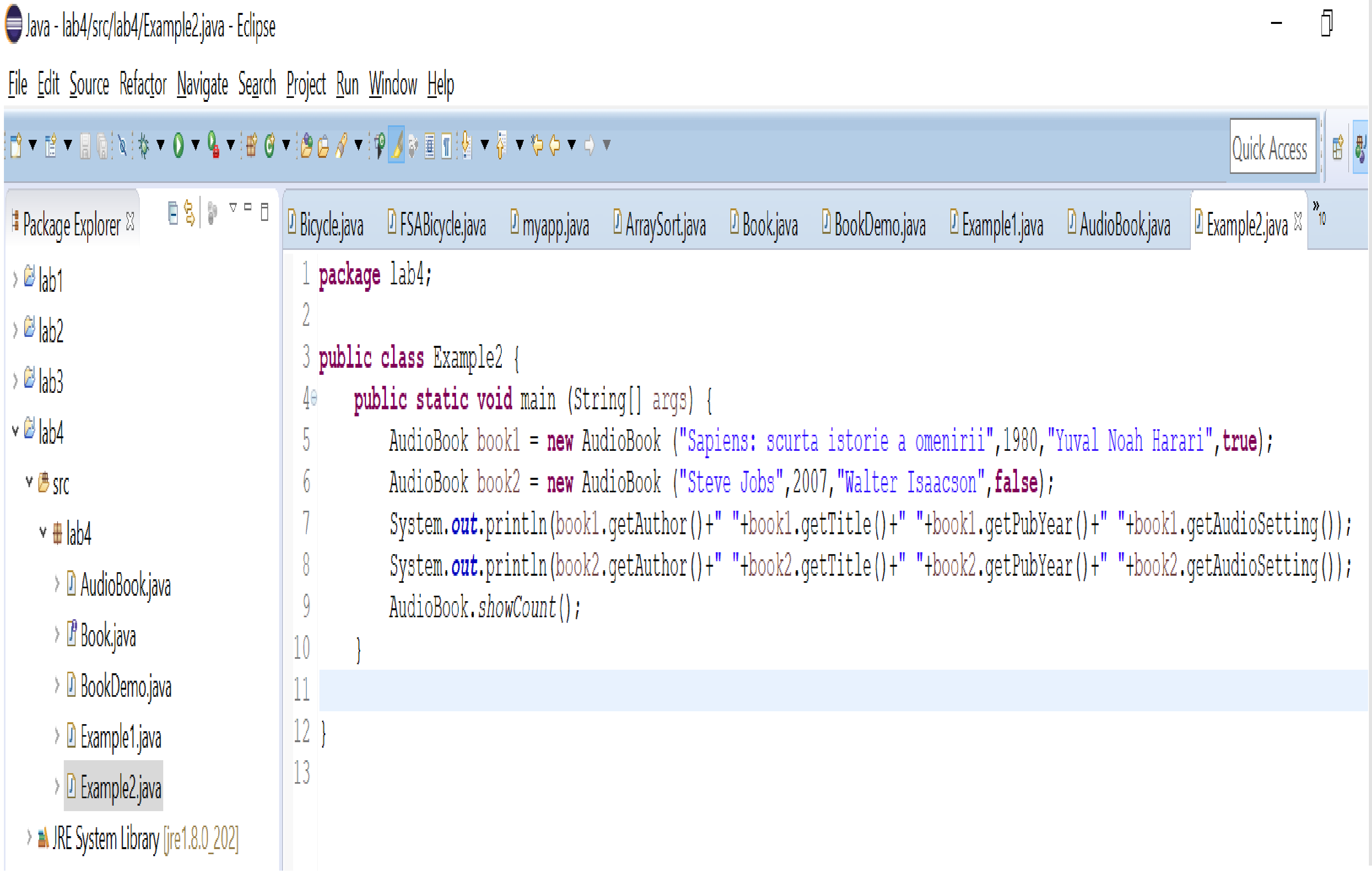
```java
package lab4;

public class AudioBook extends BookDemo {
    boolean audio;

    public Boolean getAudioSetting(){
        return this.audio;
    }

    public void setAudioSetting (Boolean _Audio){
        this.audio = _Audio;
    }

    AudioBook (String author, int year, String title, Boolean audio){
        super(title,year,author);
        this.setAudioSetting(audio);
    }

}
```

Exercise1



```java
package lab4;

public class Example1 {
    public static void main (String[] args){
        BookDemo book1 = new BookDemo();
        book1.setTitle("Sapiens: scurta istorie a omenirii");
        book1.setAuthor("Yuval Noah Harari");
        book1.setPubYear(1980);
        BookDemo book2 = new BookDemo ("Steve Jobs",2007,"Walter Isaacson");
        System.out.println(book1.getAuthor()+" "+book1.getTitle()+" "+book1.getPubYear());
        System.out.println(book2.getAuthor()+" "+book2.getTitle()+" "+book2.getPubYear());
        BookDemo.showCount();
    }

}

```

Exercise2



```java
package lab4;

public class Example2 {
    public static void main (String[] args) {
        AudioBook book1 = new AudioBook ("Sapiens: scurta istorie a omenirii",1980,"Yuval Noah Harari",true);
        AudioBook book2 = new AudioBook ("Steve Jobs",2007,"Walter Isaacson",false);
        System.out.println(book1.getAuthor()+" "+book1.getTitle()+" "+book1.getPubYear()+" "+book1.getAudioSetting());
        System.out.println(book2.getAuthor()+" "+book2.getTitle()+" "+book2.getPubYear()+" "+book2.getAudioSetting());
        AudioBook.showCount();
    }

}
```